REMARKS/ARGUMENTS

All Claims 1-29 stand rejected.

Specifically, Claims 1-10 were rejected for being directed to non-statutory subject matter. Claims 1-10 were also rejected for being obvious over some combination of three articles, namely by Lee, Nicola and Joshi and two patents, namely by Schofield and Frank.

Claims 1-10 have been canceled. New Claims 30-39 have been added. Hence, reconsideration is respectfully requested.

New Claim 30 is supported by originally-filed Claim 1 as well as throughout the originally-filed specification, including, for example, page 2, line 11 ("by way of example, a database system"), page 2 at line 12 ("allowable recovery time within which the system is to recover should a system failure…"), page 1 at lines 18-19 ("a recovery time within which the system is to recover from a system crash"), page 3 at lines 7-8 ("providing a simulated check point queue"), page 3 at line 15 ("a checkpoint queue used in normal operation"), page 2, lines 9-10 ("simulate the effects … on the runtime performance…), page 12 lines 26-28 ("using actual database transactions that are requested and which occur within the database system under normal operating conditions"), and page 2, lines 14-15 ("simulated under normal operating conditions using actual runtime data").

New Claim 30 is believed to be directed to statutory subject matter, because it is directed to a method implemented in a computer.

New Claim 30 (as well as originally-filed Claim 1) distinguishes over the combined teachings of Lee, Nicola and Joshi for a number of reasons.

Applicants submit that none of the three articles by Lee, Nicola and Joshi discloses or suggests simulating the effect of a setting for recovery time, on runtime performance of a database system as recited in new Claim 30.

In rejecting Claim 1 the Examiner relied on Nicola's statement that "increasing the frequency of checkpointing decreases recovery time but increases the checkpointing time …" (see the Office Action at page 4, bottom 4 lines). However, Nicola's article is directed to applying this just-described principle, to determine an optimum checkpointing policy, which optimizes a certain performance measure (see Nicola's Abstract at lines 3-6). Nicola further states "The purpose of this paper is to study and compare different models in order to select one which adequately represents a realistic system and yet tractable for analysis ..." (see Nicola's Abstract at lines 16-18). Nicola may at most teach how to determine recovery times for corresponding models of checkpointing. See Nicola's article, page 817, column 2, third paragraph "appropriate variables are updated (e.g. the total checkpointing or recovery time during a simulation run...)". Hence, Nicola's recovery times appear to be outputs of his simulation.

There appears to be no suggestion by Nicola (or by any of the remaining four Examiner-cited references) to do the opposite, i.e. with a setting of recovery time as input, to simulate its effect on runtime performance. Claim 30 now recites "wherein the simulated checkpoint queue is associated with a setting for recovery time whose effect on runtime performance of the database system is being simulated in said computer".

In rejecting Claim 1, the Examiner cited to Joshi's article at page 665, column 2, paragraph 3 and page 667, column 2, paragraph 3 for teaching a simulated checkpoint queue (see bottom half of page 5 of Office Action). It is unclear from this citation, as to which of two data structures described by Joshi (namely ACQ and BCQ) is the Examiner analogizing to Applicants' simulated checkpoint queue. Both paragraphs cited by the Examiner are reproduced below (with original emphasis in italics):

> In order to address these issues, the checkpointing algorithm has been completely rewritten in Oracle 8.0. Our objective was to make the algorithm scalable to very large buffer caches, and to facilitate frequent checkpointing for fast crash recovery. Scalability is achieved by organizing all the modified buffers on ordered queues; such queues increase the efficiency of identifying the precise set of buffers that need to be written for checkpoints. Frequent advancement of the database checkpoint is made possible by the introduction of incremental checkpointing.
>
> The incremental checkpoint technique uses the same data structures that are used by conventional checkpoints. It exploits the fact that the dirty buffers in the cache are linked in low RBA order. If the DBWR continually writes buffers from the head of the checkpoint queue, the instance checkpoint (lowest low-RBA of the modified buffers) will keep advancing. Periodically, CKPT can record this lowest low-RBA to the control file (using a very lightweight control-file update protocol). This periodically recorded lowest low RBA is the current position of the incremental checkpoint for the instance. Since the incremental checkpoint is performed continuously, the value of the incremental checkpoint

> RBA will be much closer to the tail of the log than the RBA of a "conventional" checkpoint, thus ...

It is also unclear from the Office Action, whether or not in the Examiner's proposed prior art combination, Joshi's queues are to be implemented as actual queues in a database system or if Joshi's queues are to be simulated as per the teachings of Lee et al.

Moreover, the Examiner appears to have misunderstood the teachings of Joshi, as evident from the following statement in the bottom half of page 6 of the Office Action (emphasis added): "Joshi ... teaches in response to ... because each such entry on the ***checkpointing queue*** contains the entry up to which the buffers need to be written in order to complete the checkpoint represented by the entry (Page 667, CL1, Para3). " A review of the cited text in Joshi's article shows that Joshi is referring to ACQ ("Each such entry on the ACQ contains the RBA up to which buffers need to be written in order to complete the checkpoint represented by the entry").

Applicants submit that Joshi's ACQ cannot be replaced by the checkpointing queue, as stated by the Examiner. Joshi's ACQ is a queue of requests for checkpoints. See Joshi's page 667, column 1, paragraph 3 ("The ACQ is a single queue that contains active checkpoint requests. When a checkpoint is requested, a new active checkpoint entry describing the request is added to the ACQ"). Joshi further states that such requests are used to change a checkpoint queue (see bottom of the second column on page 666).

Note further that Joshi's ACQ may be at most understood as modifying Joshi's BCQ, but there is no suggestion to do the opposite. Specifically, there appears to be no suggestion to respond to changes in Joshi's BCQ by modifying another queue. Hence there is no suggestion to modify a simulated checkpoint queue (which is nowhere disclosed or suggested in any of the three articles).

Applicants submit that the combined teachings of Lee, Nicola and Joshi fail to disclose or suggest a method that uses both ***a simulated checkpoint queue and a normal checkpoint queue*** (which is already present in a database system). Moreover, the combined teachings of Lee, Nicola and Joshi further fail to disclose or suggest responding to a change in a normal checkpoint queue (caused by actual database transactions), by changing the simulated checkpoint queue.

Thus, Applicants respectfully submit that new Claim 30 is patentable over the combination of Lee, Nicola and Joshi. Claims 31-39 depend from Claim 30 and are, therefore, likewise patentable.

The remaining claims 11-29 are also believed to be patentable over the prior art for one or more reasons of the type discussed above. Hence, Applicants respectfully request allowance of all pending claims.

If after reviewing the above remarks, the Examiner's position remains that the pending claims are not in allowable form, then Applicants respectfully request another non-final Office Action, due to the above-noted defects in the current Office Action. In the new Office Action, the Examiner is requested to (1) state whether or not Joshi's algorithm is to be simulated and (2) identify which of Joshi's two data structures ACQ and BCQ is analogized to Applicants' simulated checkpoint queue.

The Examiner is respectfully requested to consider all information being cited in an Information Disclosure Statement (IDS) that is being concurrently filed herewith. In particular, the Examiner is requested to review the prosecution histories of the US Patent Applications being cited, in view of the Dayco case.

Should the Examiner have any questions concerning this response, the Examiner is invited to call the undersigned at (408) 982-8203.

Via Express Mail Label No.
EV 581 855 955 US

Respectfully submitted,
S. Omkar
Omkar K. Suryadevara
Attorney for Applicants
Reg. No. 36,320

SILICON VALLEY
ATENT GROUP LLP

0 Mission College Blv
Suite 360
anta Clara, CA 95054
(408) 982-8200
FAX (408) 982-8210